

Bitwise Operation (1A)

Copyright (c) 2010-2013 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

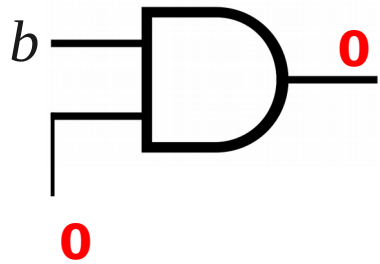
This document was produced by using OpenOffice.

Bitwise Operators

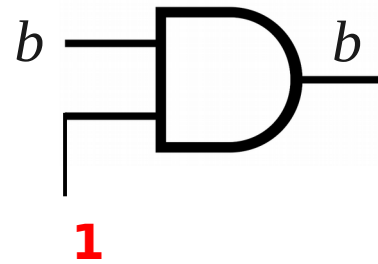
Logical AND	&&
Logical OR	
Logical XOR	n/a
Logical NOT	!

bitwise AND	&
bitwise OR	
bitwise XOR	^
bitwise NOT	~

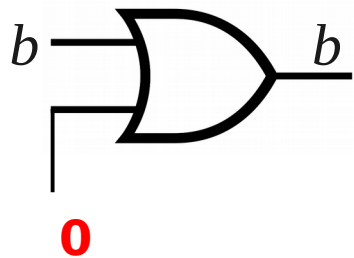
Logic Gates and Bit Operators



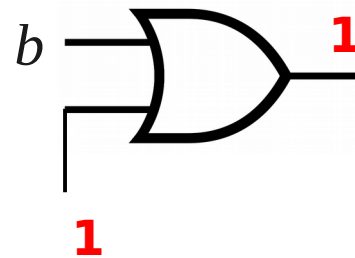
$$b \& 0 = 0$$



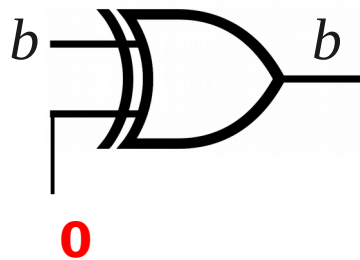
$$b \& 1 = b$$



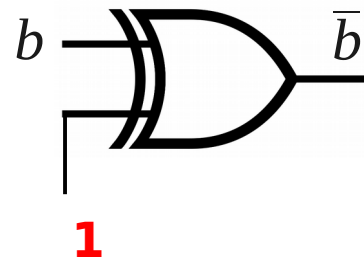
$$b | 0 = b$$



$$b | 1 = 1$$



$$b \wedge 0 = b$$



$$b \wedge 1 = \bar{b}$$

Testing Bits

```
if (x & 64) != 0)
```

```
if (x & 0x0040)
```

```
if (x & (1 << 6))
```

testing bit 6 of x

0x0040 : 64 in hex

$1 \ll 6 : 2^6 = 64$

Setting & Clearing Bits

Setting bit 7 of x

$x = x \mid (1 \ll 7)$

$x \mid= (1 \ll 7)$

Clearing bit 7 of x

$x = x \& \sim(1 \ll 7)$

$x \&= \sim(1 \ll 7)$

Inverting bit 7 of x

$x = x \wedge (1 \ll 7)$

$x \wedge= (1 \ll 7)$

0x0080 : 64 in hex
 $1 \ll 7 : 2^7 = 128$

Max Shifts for 32-bit and 64-bit numbers

32-bit shift

$1 \ll 30$ signed 2^{30}

$1U \ll 31$ unsigned 2^{31}

64-bit shift

$1L \ll 62$ signed 2^{62}

$1UL \ll 63$ unsigned 2^{63}

32-bit shift

$1 \ll 31$ (overflow)

$1U \ll 32$ (overflow)

64-bit shift

$1L \ll 63$ (overflow)

$1UL \ll 64$ (overflow)

Max Numbers for 32-bit and 64-bit numbers

32-bit shift

$$((1U \ll 31) - 1) + (1U \ll 31) \quad 2^{32} - 1$$

64-bit shift

$$((1UL \ll 63) - 1) + (1UL \ll 63) \quad 2^{63} - 1$$

Difference?

$1 \ll 32$

$1 \ll 32L$

$1 \ll 16$

$1 \ll 16L$

References

- [1] Essential C, Nick Parlante
- [2] Efficient C Programming, Mark A. Weiss
- [3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
- [4] C Language Express, I. K. Chun
- [5] “A Whirlwind Tutorial on Creating Really Teensy ELF Executables for Linux”
<http://cseweb.ucsd.edu/~ricko/CSE131/teensyELF.htm>
- [6] “Fundamentals of Embedded Software ...”, D.L. Lewis